2004

Error Analysis of Two Types of Grammar for the Purpose of Automatic Rule Refinement

Ariadna Font Llitjós
Carnegie Mellon University

Katharina Probst

Jaime G. Carbonell
Carnegie Mellon University

Error Analysis of Two Types of Grammar for the Purpose of Automatic Rule Refinement

Ariadna Font Llitjós, Katharina Probst and Jaime Carbonell

Language Technologies Institute
Carnegie Mellon University
{aria,kathrin,jgc}@cs.cmu.edu

Abstract. This paper compares a manually written MT grammar and a grammar learned automatically from an English-Spanish elicitation corpus with the ultimate purpose of automatically refining the translation rules. The experiment described here shows that the kind of automatic refinement operations required to correct a translation not only varies depending on the type of error, but also on the type of grammar. This paper describes the two types of grammars and gives a detailed error analysis of their output, indicating what kinds of refinements are required in each case.[1]

1 Introduction and Motivation

Due to the recent spread of electronic information to minority populations and their languages, the interest of MT has expanded to language pairs that have previously been untackled because of the lack of large parallel corpora and the lack of human experts that could design translation grammars.

This poses a new challenge to the MT community, and requires to research different MT approaches which, given very little data, are able to build a working MT system and, most importantly, are able to improve this initially small system in a semi-automatic way, not requiring either computer experts or linguists that know the language.

Our approach is to carefully elicit user feedback about MT system output correctness, and use it to automatically refine the translation grammar and lexicon as required to accommodate the changes made by users. Translation grammars in our transfer-based MT system can be both hand-crafted and automatically learned, thus it is interesting to see what are the differences between them in terms of coverage, parsimony, ambiguity, error rate and translation quality.

This paper describes an experiment with these two types of grammars designed to explore whether the automatic rule refinement operations that need to be applied in each case are going to be different and, if so, how.

After a brief description of our MT system focusing on the Rule Learning and Rule Refinement modules, the two grammars used in the experiment are described in detail and the error analysis indicating what kinds of refinements are required in each case follows.

[1] This research was funded in part by NSF grant number IIS-0121631.

2 MT Approach

Our MT research group at Carnegie Mellon has been working on a new MT approach, under the AVENUE project, that is specifically designed to enable rapid development of MT for languages with limited amounts of online resources. Our approach assumes the availability of a small number of bilingual speakers of the two languages, but these need not be linguistic experts. The bilingual speakers create a comparatively small corpus of word aligned phrases and sentences (on the order of magnitude of a few dozen to a few thousand sentence pairs) using a specially designed elicitation tool [7].

From this data, the learning module of our system automatically infers hierarchical syntactic transfer rules, which encode how constituent structures in the source language (SL) transfer to the target language (TL). The collection of transfer rules, which constitute the translation grammar, is then used in our run-time system to translate previously unseen SL text into the TL [6].

The AVENUE transfer-based MT system consists of four main modules: elicitation of a word aligned parallel corpus; automatic learning of translation rules; the run time transfer system, and the interactive and automatic refinement of the translation rules.

2.1 The Transfer Rule Formalism

In our system, translation rules have 6 components: a) the type information, which in most cases corresponds to a syntactic constituent type; b) part-of speech/constituent sequence for both the SL and the TL; c) alignments between the SL constituents and the TL constituents; d) `x`-side constraints, which provide information about features and their values in the SL sentence; e) `y`-side constraints, which provide information about features and their values in the TL sentence, and f) `xy`-constraints, which provide information about which feature values transfer from the source into the target language.

For illustration purposes, Figure 1 shows an example of an English to Spanish translation rule for noun-phrases containing a noun and an adjective. This translation rule swaps the original English word order, from adjective-noun to noun-adjective, enforces their agreement in Spanish and ensures that the noun is the head of the NP. The feature unification equations used in the rules follow a typical unification grammar formalism. For more details about the rule formalism, see [6].

3 Learning Rules

For the experiment described in this paper, the translation direction is English to Spanish, and the latter is used for illustration purposes to simulate a resource-poor language. In the following discussion of the learning procedure, the x-side always refers to English, whereas the y-side refers to Spanish.

```
{NP,9}  ;;; Rule identifier ;;;
   NP::NP : [ADJ N] -> [N ADJ]
   ((x1::y2) (x2::y1) ; set constituent alignments
   ((x0 mod) = x1) ; the adjective is the modifier
   (x0 = x2)   ; the noun is the head
   (y2 == (y0 mod)) ; building Spanish sentence f-structure
   (y1 = y0)
   ((y2 agr) = (y1 agr)))
```

Fig. 1. Sample Translation Rule. x means source (here: English) and y, target (here: Spanish).

The first step in the Rule Learning (RL) module is Seed Generation. For each training example, the algorithm constructs at least one 'seed rule', i.e. a flat rule that incorporates all the information known about this training example, producing a first approximation to a valid transfer rule. The transfer rule parts can be extracted from the available information or are projected from the English side.

After producing the seed rule, compositionality is added to it. Compositionality aims at learning rules that can combine to cover more unseen examples. The algorithm makes use of previously learned rules. For a given training example, the algorithm traverses through the parse of the English sentence or phrase. For each node, the system checks if there exists a lower-level rule that can be used to correctly translate the words in the subtree. If this is the case, then an element of compositionality is introduced.

In order to ensure that the compositionality algorithm has already learned lower-level rules when it tries to learn compositional rules that use them, we learn rules for simpler constituents first. Currently, the following order is applied: adverb phrases, adjective phrases, noun phrases, prepositional phrases, and sentences. While this improves the availability of rules for the compositionality module at the right time, the issue of co-embedding still needs to be resolved. For more details, see [6].

4 Refining Rules

Instead of having to pay a human translator to correct all the sentences output by an MT system, we propose the use of bilingual speakers to obtain information about translation errors and use this information to correct the problems at their root. Namely, by refining the translation rules that generated the errors.

The refinement process starts with an interactive step, which elicits information about the correctness of the translations from users with the Translation Correction Tool (TCTool). This tool allows bilingual users to correct the output of an MT system and to indicate the source of the error. Users can insert words, change the order of words, or specify that an agreement was violated.

To find more about the reliability of translation correction feedback given by bilingual speakers, we ran an English to Spanish user study using the manually written grammar described in 5.1. We found that users can detect translation errors with reasonably high accuracy (89%), but have a harder time determining what type of error it is. Given our MT error classification, users identified the error correctly 72% of the time. For more details about the TCTool and the English-Spanish user study, see [3].

The second step of the refinement process is to do the actual refinement. The Rule Refinement (RR) module modifies the translation grammar according to what user feedback suggests.

The first goal of the RR module is to search the feature space to determine which set of features triggered a particular correction. To do this, the RR module needs to rely on active learning methods to try to discover the underlying principles and constraints of the TL language grammar, which were not elicited at the first stage of the development process, or which were not learned by the RL module.

For RR purposes, we define the difference between two TL minimal pair sentences as the intersection of the set of feature attributes for which they have different values, not taking into consideration the user correction we are seeking to account for. We call this the feature delta function (δ) and, in the general case, it can be written as follows:

$$\delta(\mathrm{T1},\mathrm{T2}) = \bigcap_{i=1}^{n} \delta(\mathrm{wT1}_i,\mathrm{wT2}_i) \tag{1}$$

for all the *relevant* words that are different in T1 and T2 ($\mathrm{wT1}_i \neq \mathrm{wT2}_i$) and where `n` is the length of the sentence.

The resulting delta set is used as a guidance to explore the feature space of potentially relevant attributes, until the RR module finds the ones that triggered the correction, and can add the appropriate constraints to the relevant rules.

Suppose we are given the English sentence `Juan is a great friend`, and the translation grammar has the following general rule for noun phrases `NP::NP : [ADJ N] -> [N ADJ]` (see Figure 1 for the complete rule), then the translation output from the MT system will be `Juan es un amigo` ***`grande`*** (T1), since that is what the general rule for nouns and adjectives in Spanish dictates. However, this sentence instantiates an exception to the general rule, and thus, the user will most likely correct the sentence into `Juan es un` ***`gran`*** `amigo` (T1').

Now, the RR module needs to find a minimal pair (T2) that illustrates the linguistic phenomenon that accounts for the MT error. Let's assume that `Juan is a smart friend` is also in the elicitation corpus and it gets correctly translated as `Juan es un amigo inteligente`.

The delta function of these two TL sentences evaluated is shown below:

$$\delta(\texttt{(Juan es un amigo grande),(Juan es un amigo inteligente)}) = \\ = \delta(\texttt{grande,inteligente}) = \{\}$$

Since we are comparing minimal pairs, the delta function is reduced to comparing just words that are different between the two TL sentences but are aligned to the same SL word, or are in the same position and have the same POS, and we do not need to calculate the delta functions for differing words that are not relevant, such as δ(un,amigo).

If the delta set had contained one feature attribute that, when changed in T2, users would correct it in the same way they corrected T1, then we would hypothesize that it accounts for the correction and would bifurcate the NP rule based on that attribute.

Since both adjectives are singular and undefined with respect to gender, the delta set is empty and thus the RR module determines that the existing feature set is insufficient to explain the difference between prenominal and postnominal adjectives, and therefore it postulates a new binary feature, `feat_1`.

Once the RR module has determined what are the triggering features, it proceeds to refine the relevant grammar and lexical rules to include prenominal adjectives by adding the appropriate feature constraints. In this case, the RR module creates a duplicate of the NP rule shown in Figure 1, switches `N ADJ` to `ADJ N` on the target side and adds the the following constraint to it: `(feat_1 = +)`. The original rule also needs to be refined to include the same constraint with the other feature value `(feat_1 = -)`. The lexicon will later need to code for the new feature as well.

More detailed examples of minimal pair comparison and the feature space exploration algorithm can be found in [2].

5 English-Spanish Experiment

We ran an experiment with two different grammars, a manually written one and an automatically learned one using the RL module described in section 3.

The training set contains the first 200 sentences of the AVENUE elicitation corpus, and the test set contains 32 sentences drawn from the next 200 sentences in the corpus (the same that was used for the English-Spanish user study [3]). Both MT systems used a lexicon with 442 entries, developed semi-automatically seeking to cover the training and the test sets (400 sentences), so that the effectiveness of rules can be measured abstracting away from lexical gaps in the translation system. The description of the two types of grammar follow.

5.1 Hand-crafted Translation Grammar

The manually written grammar contains 12 translation rules: 2 S rules, 7 NP rules, 3 VP rules. It took a grammar writer about two weeks to develop and debug the grammar. Examples of translation rules in this grammar are given below.

```
{S,0} ;;; Rule identifier ;;;
  S::S : [NP VP] -> [VP]
```

```
{VP,3}
  VP::VP : [VP NP] -> [VP NP]
```

```
; x0 y0   x1 x2      y1
((X2::Y1) ; set constituent alignments
  ((x1 agr pers) = (*OR* 1 2))
  ((x1 agr num) = sg)
  ((x1 case) = nom)
  (x0 = x2)
  ((y1 agr) = (x1 agr)))
```

```
((X1::Y1)  (X2::Y2)
  ((x2 case) = acc)
  ((x0 obj) = x2)
  ((x0 agr) = (x1 agr))
  (y2 == (y0 obj))
  ((y0 tense) = (x0 tense))
  ((y0 agr) = (y1 agr)))
```

Spanish is a pro-drop language, so the S rule above causes the subject to be dropped when the subject NP is a 1st or 2nd singular personal pronoun (since this is what the translator of the elicitation corpus did), and it also ensures agreement between the Spanish verb and the English subject. The VP rule illustrates the common syntactic structure were the direct object follows the verb, both in English and Spanish. Figure 1 shows an example of a manually written NP rule.

5.2 Automatically Learned Translation Grammar

The RL module described in section 3 can produce a basic and an enhanced grammar fully automatically in about a minute, as it is exactly learned from the elicitation corpus examples [6]. In general, the automatic learning method described in this paper is linear in the input size.

The enhanced version essentially expands the training corpus by traversing the English parse tree from the top down. For each internal node, the enhancement algorithm extracts the subtree rooted at this node. Then, using the word alignments provided by the bilingual user, it forms a new training example for the English and Spanish chunks covered by the subtree. The rule learning algorithm can then be run as is on the expanded training set, resulting in a larger grammar.

For the present experiment, the English training data was parsed using the Charniak parser [1]. A few feature constraints extracted from morphological analyzers for both English and Spanish (gender, number, person and tense) where input to the Rule Learner and the final grammar was able to learn some feature constraints. The enhanced version of the grammar with constraints contains 316 translation rules (194 S, 43 NP, 78 VP and 1 PP), 223 more than the learned grammar with no constraints.

Examples from the final learned grammar (enhanced and with features) are shown below.

```
{S,90}
S::S [NP VP] -> [NP VP]
((X1::Y1)(X2::Y2)
((X1 def) = +)
((X2 def) = -)
((X2 agr num) = (X1 agr num))
((X2 tense) = past)
((Y2 agr gen) = (Y1 agr gen))
((Y2 agr num) = (Y1 agr num)))
```

```
{VP,46}
VP::VP [V NP] -> [V NP]
((X1::Y1)(X2::Y2)
((X2 def) = -)
((X2 agr num) = sg)
((Y1 agr pers) = 2)
((Y2 agr gen) = masc)
((Y2 agr num) = (Y1 agr num)))
```

Overall, the enhanced grammar rules learned by the Rule Learner make useful generalizations and are quite close to what a linguist might write, however they often contain constraints that are either too specific or too general. In the rules above, the constraint about the tense of the verb in `S,90` is too specific; ideally we would like this rule to apply to all VPs regardless of their tense.

On the other hand, the Rule Learner can overgeneralize; when it finds that a feature value match occurs often enough, it will decide that there is a correlation, and thus an agreement constraint gets added to the rule. This is illustrated by the number agreement constraint in rule `VP,46`. Because enough examples in the training data had the verb and the object coincide in number, it assumes that a generalization can be made. In Spanish, however, verbs and their objects do not always agree.

5.3 Grammar Ambiguity

The reason there are many more rules in the learned grammar is that the current implementation of the RL module does not throw away rules that are too specific and that are subsumed by rules which have achieved a higher level of generalization during the learning process.

When running our transfer system on a test set of 32 sentences, it was observed that the manually written grammar results in less ambiguity (on average 1.6 different translations per sentence) than the automatically learned grammar (18.6 different translations per sentence). At the same time, the final version of the learned grammar results in less ambiguity than the learned grammar with no constraints. This is to be expected, since relevant constraints will restrict the application of general rules to the appropriate cases.

Additional ambiguity is not necessarily a problem; however, the goal of the transfer engine should be to produce the most likely outputs first, i.e. with the highest rank. In experiments reported elsewhere [4], we have used a statistical decoder with a TL language model to disambiguate between different translation hypotheses. We are currently investigating methods to prioritize rules and partial analyses within the transfer engine, so that we can rank translation hypotheses also when no TL language model is available.

While this work is under investigation, we emulated this module with a simple reordering of the grammar: we reordered three rules (2 NPs and 1 S rule) that had a high level of generalization (namely containing agreement constraints instead of the more specific value constraints) to be at the beginning of the grammar. This in effect gives higher priority to the translations produced with these rules.

5.4 Comparing Grammar Output

Most of the translation errors produced by the manual grammar can be classified into lack of subj-pred agreement, wrong word order of object pronouns (clitic), wrong preposition and wrong form (case) and out-of-vocabulary word. On top of the errors produced by the manual grammar, the current version of the learned

grammar also had errors of the following types: missing agreement constraints, missing preposition and overgeneralization.

An example of differences between the errors produced by the two types of grammar can be seen in the translation of `John and Mary fell`. The manual grammar translates it as `*Juan y Maria cayeron`, whereas the learned grammar translated it as `**Juan y Maria cai`. The learned grammar does have an NP rule that covers `[Juan y Maria]`, however it lacks the number constraint that indicates that the number of an NP with this constituent sequence (`[NP CONJ NP]`) has to be plural. The translation produced by the manual grammar, `*Juan y Maria cayeron`, is also ungrammatical, but the translation error in this case is a bit more subtle and thus much harder to detect. The correct translation is `Juan y Maria` *`se`* `cayeron`.

Another example to illustrate this is the translations of `John held me with his arm`. The MT system with the manual grammar outputs `*Juan sujetó me con su brazo`, whereas the one with the learned grammar outputs `**A Juan sujetó me con su brazo`.

Sometimes the output from the learned grammar is actually better than the manual grammar output. An example of this can be seen in the translations of `Mother, can you help us?`. The manual grammar translated this as `**Madre, puedo ayudar nos?` and the learned grammar translates is as `*Madre, puedes tu ayudar nos?`. The number of corrections required to fix both translations is the same, but the one produced by the learned grammar is clearly better.

5.5 Error Analysis

The 32 sentences used in the English-Spanish user study described in [3] were translated using both grammars described above. We looked at the 5 first translations only, and found that both grammars translated the same 4 SL sentences correctly. Even though the final number of SL sentences correctly translated by the two grammars is the same, overall, the quality of the translations produced by the learned grammar was worse.

We looked at the best translation for each source language sentence in the user study for one grammar and compared it with the best translation for the other grammar. The results can be seen in Table 1 below.

Table 1. Error analysis comparing a learned grammar with a manually written grammar. See Section 6 for examples.

	number of sentences (over 32)
same translation	17 (3 correct)
manual grammar better	10
learned grammar better	2
different error type	3 (1 correct)

6 Rule Refinements Required for Each Type of Grammar

Since 15 out of 32 sentences contained different translation errors for each grammar, the refinement operations required by the translations rules of one grammar and by the other grammar might not be the same for a given SL sentence. Following are a few examples of the cases listed in Table 1.

Same translation

In many cases, the sentences were translated in the same way by the two grammars, and thus they need the same refinements to be corrected. Sentences that only differ in the optional subject pronoun are included in this category. These are two examples from the test set that illustrate this:

SL: I saw you yesterday - TL: *(Yo) vi *tu* ayer

SL: It was a small ball - TL: eso era un balón pequeño

Different Translation

For 10 sentences in the test set, even though both translations are wrong, the manual grammar output requires less changes to be correct:

SL: I'm proud of you - TL-l: **Yo estoy orgulloso *tu* - TL-m: *Yo estoy orgulloso de *tu*

besides changing the `tu` into `ti` in both cases, the learned grammar needs an extra refinement, the addition of "de" in front of the second person pronoun.

In addition to the example given section 5.4, the learned grammar output requires less amount of refinement for the following SL sentence:

SL: I like you - TL-l: **Yo* me gustas tu - TL-m: **Me *gusta* tu

In a few cases, both grammars output equally bad translations, but with different kinds of errors:

SL: He looked at me - TL-l: *él *miraron* me - TL-m: * él miró *en me*

The current experiment reveals a couple of main interesting differences between the refinements required by hand-crafted grammars and automatically learned grammars. Several rule refinements required to correct sentences translated by the hand-crafted grammar involve bifurcating a rule to encode an exception, whereas a larger portion of the refinements necessary to refine the learned grammar involve adding or deleting agreement constraints. More details on the refinement operations can be found in [2].

In general, the biggest differences between the hand-crafted and the learned grammar is that the learned grammar has a higher level of lexicalization and, currently, it can make good use of the Rule Refinement module to make adjustments to the feature and agreement constraints to achieve the appropriate level of generalization.

7 Conclusions and Future Work

The Rule Refinement module, together with the Translation Correction Tool, can improve both hand-crafted and automatically learned translation grammars.

The addition of feature constraints to the automatically learned grammar brings the two types of grammar and their output closer. However, with the

current implementation of the learned grammar, the Rule Refinement module will still give the most leverage when combined with an automatically learned grammar.

Given translations for the same SL sentence, both the types of translation rules and the kind of errors they produce differ almost 50% of the time depending on the type of grammar, thus the Rule Refinement module will need to detect the difference in the translation rules and perform the appropriate refinement operations, accordingly.

In general, manually-written grammar rules will need to be refined to encode exceptions, whereas automatically-learned grammars will need to be refined so that they achieve the right level of generalization.

We would like to do some experiments to check whether using a user-corrected pair as new training data for the Rule Learning module and using the user feedback to refine the rules with the Rule Refinement module yield identical or similar results, and what is the best way to improve translation quality over a corpus, once there is an initial grammar, which can be either manually written or automatically learned.

Finally, we would also like to look into using reference translations created to evaluate MT system output [5] to improve our grammar and lexicon automatically. We might be able to do this by applying some minimal edited distance metric, and running a morphological analyzer for at least the target language.

References

1. Charniak, E.: A Maximum-Entropy-Inspired Parser, North American chapter of the Association for Computational Linguistics (NAACL), 2000.
2. Font Llitjós, A.: Towards Interactive and Automatic Refinement of Translation Rules, PhD Thesis Proposal, Carnegie Mellon University, forthcoming in August 2004 (`www.cs.cmu.edu/~aria/ThesisProposal.pdf`).
3. Font Llitjós, A., Carbonell, J.: The Translation Correction Tool: English-Spanish user studies, 4th International Conference on Language Resources and Evaluation (LREC), 2004.
4. Lavie, A., Vogel, S., Levin, L., Peterson, E., Probst, K, Font Llitjós, A., Reynolds, R., Carbonell, J., Cohen, R.: Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario, ACM Transactions on Asian Language Information Processing (TALIP), 2:2, 2003.
5. Papineni, K., Roukos, S., Ward, T.: Maximum Likelihood and Discriminative Training of Direct Translation Models, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-98), 1998.
6. Probst, K., Levin, L., Peterson, E., Lavie, A., Carbonell J.: MT for Resource-Poor Languages Using Elicitation-Based Learning of Syntactic Transfer Rules, Machine Translation, Special Issue on Embedded MT, 2003.
7. Probst, K., Brown, R., Carbonell, J., Lavie, A., Levin, L., Peterson, E.: Design and Implementation of Controlled Elicitation for Machine Translation of Low-density Languages, Workshop MT2010 at Machine Translation Summit VIII, 2001.